**类私有属性：**

例如类A 有实例私有属性\_\_name,

在类外部，我们不能修改次属性，但当我们尝试修改此属性时，并不会报错，而是另外生成一个该属性，例如：

class A:

def \_\_init\_\_(self,value):

self.\_\_name=value

def inf(self):

print(‘private variable:’,self.\_\_name)

a=A(3)

a.inf()

a.\_\_name=2

**a.inf()**

**print(a.\_\_name)**

注意后两行输出不一样，前者是3，后者是2（新开辟的变量）。

但是self.\_name事实上并不是私有变量，依然可以在外部修改。

**Lambda 在闭包中会保存局部变量**

my\_list=[lambda: i for i in range(5)]

for l in my\_list:

print(l())

结果：4，4，4, 4,4

def groupby(items,size):

return zip[\*[iter(items)]\*size]

groupby(range(9),3) => [(0,1,2),(3,4,5),(6,7,8)]

**访问类的私有方法或变量**

**1:** 可以 \_+类名字+私有方法名字，例如

a.\_A\_\_privatemethodname(),通过实例a 访问私有属性

**2:** 可以在类中定义一个公有的接口，实例通过调用该接口见解地调用私有方法或属性。

**else 语句的相关说明：**

1:在异常处理时，当try 语句块中没有异常，就会执行else 的语句

2: 在while,for 等语句中也支持else 语句，含义是当其后面的条件执行完后（for） 或者后面条件不成立后（while），最后执行else 语句。但是当有break 语句提前跳出loop时，else 语句就不会执行。

例如：

i＝0

while i<2:

i+=1

print i

break

else:

print(‘complete while loop’)

=> 1 ,因为break 跳出循环了。

**元类：**

python中所有的类都是type的实例，因此type是该元类，当然也可以自定义自己的元类，需要重写\_\_init\_\_ 和\_\_new\_\_方法，在调用该类时，2.x 版本需要在类内定义一个\_\_metaclass\_\_ 属性为定义的元类名如Mymeta，在3.x版本中在定义类时直接：

class(metaclass=Mymeta) 即可。

**关于遍历嵌套list：**

>>> a = [1, 2, [3, 4], [[5, 6], [7, 8]]]

>>> flatten = lambda x: [y for l in x for y in flatten(l)] if type(x) is list else [x]

>>> flatten(a)

[1, 2, 3, 4, 5, 6, 7, 8]

**关于collections 包：**

**namedtuple:**

>>> Point = collections.namedtuple('Point', ['x', 'y'])

>>> p = Point(x=1.0, y=2.0)

>>> p

Point(x=1.0, y=2.0)

>>> p.x

1.0

>>> p.y

2.0

但是不能修改，如p.x=2不行。

**counter**:统计一个可迭代对象出现频率较高的元素，可以适用与+,-等类似与集合的操作，并且可以调用A.most\_common(3) 返回频率为前三高的元素。

**OrderedDict**:比如一般的字典：

>>> m = dict((str(x), x) for x in range(10))

>>> print ', '.join(m.keys())

1, 0, 3, 2, 5, 4, 7, 6, 9, 8

是无序的。

可以调用可排序的字典，

>>> m = collections.OrderedDict((str(x), x) for x in range(10))

>>> print ', '.join(m.keys())

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

**defaultdict:**自定义字典value的类型。

一般的字典不可以，比如：

>>> m = dict()

>>> m['a']

Traceback (most recent call last):

  File "<stdin>", line 1, in <module>

KeyError: 'a'

会报错。

但是defaultdict会初始化类型。如：

 m = collections.defaultdict(int)

 m = collections.defaultdict(list)

甚至可以自定义自定义类型，如：

>>> m = collections.defaultdict(lambda: '[default value]')

>>> m['a']

'[default value]'

**对象到唯一计数的映射:**

|  |
| --- |
| >>> import itertools, collections  >>> value\_to\_numeric\_map = collections.defaultdict(itertools.count().next)  >>> value\_to\_numeric\_map['a']  0  >>> value\_to\_numeric\_map['b']  1  >>> value\_to\_numeric\_map['c']  2  >>> value\_to\_numeric\_map['a']  0  >>> value\_to\_numeric\_map['b']  1 |

**关于heapq:**

**返回最大的几个或最小的几个，**

heapq.nsmallest(5, a)

heapq.nlargest(5, a)

**关于多线程：**

**对象**

import threading

# mutex lock

mlock=threading.Lock()

rlock=threading.RLock()

a=1

def foo():

global a

mlock.acquire()

a+=1

print a

mlock.release()

for i in xrange(10):

tmp = threading.Thread(target=foo)

tmp.start()

**关于socket:**



例如：本地8125端口开启了一个server,返回收到的前10 个字节



（在终端输入 nc localhost 8125,就建立了tcp/ip 连接）

**关于python代码优化的一些技巧：**

<http://blog.jobbole.com/24197/>

**关于查看文件名的权限问题，os和stat包：**

将os.stat() 和stat配合使用：

s = "private.html"

st = os.stat(s)

print(bool(st.st\_mode & stat.S\_IROTH))